

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicants : Rui Li
Application No. : 10/756,152
Filed : January 12, 2004
For : METHOD AND APPARATUS FOR MAINTAINING LONGER
PERSISTENT CONNECTIONS

Examiner : Bharat Barot
Art Unit : 2155
Docket No. : 350078.413
Date : November 10, 2008

Mail Stop Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

APPELLANT'S BRIEF

Commissioner for Patents:

This brief is in furtherance of the Notice of Appeal, filed in this case on September 8, 2008. The fees required under Section 1.17(c), and any required request for extension of time for filing this brief and fees therefor, are dealt with in the accompanying papers.

I. REAL PARTY IN INTEREST

Foundry Networks, Inc. is the real party in interest.

II. RELATED APPEALS AND INTERFERENCES

None

III. STATUS OF CLAIMS

Claims 1-36 and 43-52 are pending and have been finally rejected. The rejection of claims 1-36 and 43-52 is being appealed herein. Claims 37-42 have been previously canceled.

IV. STATUS OF AMENDMENTS

There are no outstanding amendments. The last amendment/response was filed on March 5, 2008 (hereinafter referred to as “the amendment of March 5, 2008”). A final Office Action was mailed on June 12, 2008 (hereinafter referred to as “the final Office Action of June 12, 2008”) in response to the amendment of March 5, 2008 and which finally rejected claims 1-36 and 43-52.

V. SUMMARY OF CLAIMED SUBJECT MATTER

As discussed on pages 2-4 of present U.S. Application Serial No. 10/756,152 (hereinafter referred to as “the present application”), a default feature of hypertext transfer protocol (HTTP) version 1.1 is to provide a persistent connection between a client and a server. However, clients or servers may prematurely cause the (default) persistent connection to close, by for example issuing, a RESET packet, a FIN packet, a request having a “Connection: Close” header, or other communication to signal the connection to close. *See, e.g.*, page 2, line 28 to page 3, line 3, and page 4, lines 14-24 of the present application. Moreover, with earlier HTTP versions such as HTTP version 1.0, the connection is by default not maintained persistent—a HTTP 1.0 client or server needs to explicitly tell the other to keep the connection persistent, by inserting a “Connection: Keep-Alive” header in the message sent to the other side—an HTTP 1.0 client, for example, sending a request without the “Connection: Keep-Alive” header will signal the server to close the connection. *See, e.g.*, page 3, lines 4-8, and page 4, lines 25-27 of the present application.

Accordingly, one or more embodiments disclosed in the present application provides various techniques to maintain longer persistent connections. That is, to avoid the problems identified above and described in the present application, and to take more advantage of persistent connections such as those provided by the default behavior of HTTP 1.1, an embodiment treats the connection between a client and a server as two connections: a client-side connection and a server-side connection. If one side wants to close the connection, the embodiment just closes one side of the connection but leaves the other side connection open (persistent). Techniques are also provided for maintaining persistent connections in situations where an HTTP 1.0 client (wherein the default behavior is not to maintain a persistent

connection) communicates with a HTTP 1.1 server, or vice versa. Techniques are further provided for maintaining persistent connections in situations where an HTTP 1.0 client communicates with an HTTP 1.0 server, or vice versa. *See, e.g.*, page 6, lines 11-23 of the present application.

Non-exhaustive and non-limiting examples of the various techniques disclosed by the present application to maintain longer persistent connections include the following:

- de-linking the server-side connection from the closed client-side connection in response to a RESET packet sent by the client (*see, e.g.*, page 8, lines 18-28 of the present application);
- de-linking the server-side connection from the closed client-side connection in response to a FIN packet sent by the client (*see, e.g.*, page 9, lines 1-7 of the present application);
- replacing a “Connection: Close” header in the communication with a “Connection: Keep-Alive” header (*see, e.g.*, page 10, lines 3-16 of the present application);
- inserting a “Connection: Keep-Alive” header in the communication (*see, e.g.*, page 11, lines 5-10 of the present application);
- erasing/distorting/modifying a “Connection: Close” header in the communication so as to be unrecognizable (*see, e.g.*, page 11, line 16 to page 12, line 3 of the present application); and
- changing an HTTP version value indicated in the communication from a version not associated with a persistent communication (such as version 1.0) to a version associated with a persistent connection (such as version 1.1) (*see, e.g.*, page 12, lines 4-14 of the present application).

The following discusses independent claims 1, 14, 22, and 31. According to 37 CFR 41.67(c)(1)(v), a concise explanation of the subject matter in the independent claims has been set forth below with reference to the specification by page and line numbers, and to the drawings, if any, by reference characters. Accordingly, the following shows claims 1, 14, 22, and 31 together with the required reference information in brackets [] and *italicized*. Of course, the reference numbers and other bracketed information are illustrative only and are not intended to limit the claims only to the exact embodiments shown and described in the specification and figures of the present application.

1. A method, comprising:

at a network device [108 in Figure 1 and Figures 3-4; page 7, lines 8-20], providing a hypertext transfer protocol (HTTP) client-side connection and a HTTP server-side connection [page 8, lines 4-17; client-side connection line between client 102 and switch/gateway 108 in Figures 3-4; and server-side connection line 304/402 between switch/gateway 108 and server 114 in Figures 3-4];

receiving at said network device via said client-side connection a communication that signals said server-side connection to close [Request 1 at 202 in Figure 2; Request 1 at 300 in Figure 3; Request 1 at 400 in Figure 4; page 9, lines 22-26; page 10, lines 3-5; and page 12, lines 5-9]; and

maintaining persistent, by said network device, at least the server-side connection in response to said communication received via said client-side connection [page 5, lines 6-7; Request 1' in Figure 3; Request 1' at 404 in Figure 4; page 8, lines 18-28; page 9, lines 1-7; page 10, lines 3-16; page 11, lines 5-10; page 11, line 16 to page 12, line 3; and page 12, lines 4-14].

14. A method, comprising:

establishing at a network device [108 in Figure 1 and Figures 3-4; page 7, lines 8-20] a client-side connection and a server-side connection [page 8, lines 4-17; client-side connection line between client 102 and switch/gateway 108 in Figures 3-4; and server-side connection line 304/402 between switch/gateway 108 and server 114 in Figures 3-4];

reading, by said network device, content of a packet received via the client-side connection [Request 1 at 202 in Figure 2; Request 1 at 300 in Figure 3; Request 1 at 400 in Figure 4; page 9, lines 22-26; page 10, lines 3-5; page 12, lines 5-9; and page 16, lines 9-11]; and

extending, by said network device, persistency of the server-side connection if the read content of the packet signals said server-side connection to close [page 5, lines 6-7; Request 1' in Figures 3-4; page 8, lines 18-28; page 9, lines 1-7; page 10, lines 3-16; page 11, lines 5-10; page 11, line 16 to page 12, line 3; and page 12, lines 4-14].

22. An article of manufacture, comprising:

a storage medium having instructions stored thereon [112 in Figure 1; page 7, lines 11-13] that are executable by a processor [110 in Figure 1; page 7, lines 13-15] of a network device [108 in Figure 1 and Figures 3-4; page 7, lines 8-20] to:

provide at said network device a hypertext transfer protocol (HTTP) client-side connection and a HTTP server-side connection [page 8, lines 4-17; client-side connection line between client 102 and switch/gateway 108 in Figures 3-4; and server-side connection line 304/402 between switch/gateway 108 and server 114 in Figures 3-4];

receive via said client-side connection a communication that signals said server-side connection to close [Request 1 at 202 in Figure 2; Request 1 at 300 in Figure 3; Request 1 at 400 in Figure 4; page 9, lines 22-26; page 10, lines 3-5; and page 12, lines 5-9]; and

maintain persistent, by said network device, at least the server-side connection in response to said communication received via said client-side connection [page 5, lines 6-7; Request 1' in Figure 3; Request 1' at 404 in Figure 4; page 8, lines 18-28; page 9, lines 1-7; page 10, lines 3-16; page 11, lines 5-10; page 11, line 16 to page 12, line 3; and page 12, lines 4-14].

31. An apparatus, comprising:

a network device [108 in Figure 1 and Figures 3-4; page 7, lines 8-20] having:

first and second communication terminals [Figures 1 and 3-4: terminal of the switch/gateway 108 coupled to the client 102 and terminal of the switch/gateway 108 coupled to the server 114; page 20, line 11], the first terminal being associated with a hypertext transfer protocol (HTTP) client-side connection and the second terminal being associated with a HTTP server-side connection [page 8, lines 4-17; client-side connection line between client 102 and switch/gateway 108 in Figures 3-4; and server-side connection line 304/402 between switch/gateway 108 and server 114 in Figures 3-4];

a processor coupled to the first and second communication terminals [110 in Figure 1; page 7, lines 13-15; page 20, line 15]; and

software executable by the processor [112 in Figure 1; page 7, lines 11-15] to maintain persistent [page 5, lines 6-7; Request 1' in Figure 3; Request 1' at 404 in Figure 4; page

8, lines 18-28; page 9, lines 1-7; page 10, lines 3-16; page 11, lines 5-10; page 11, line 16 to page 12, line 3; and page 12, lines 4-14] at least the server-side connection in response to a communication received via said client-side connection that signals said server-side connection to close [*Request 1 at 202 in Figure 2; Request 1 at 300 in Figure 3; Request 1 at 400 in Figure 4; page 9, lines 22-26; page 10, lines 3-5; and page 12, lines 5-9*].

The following claims 27-30 and 49-51 contain means-plus-function elements falling under the scope of 35 U.S.C. § 112, sixth paragraph. Accordingly, reference information in brackets [] and *italicized* is shown in the text of claims 27-30 and 49-51 below, so as to comply with 37 CFR 41.37(c)(1)(v) which requires “For each independent claim involved in the appeal and for each dependent claim argued separately under the provisions of paragraph (c)(1)(vii) of this section, every means plus function and step plus function as permitted by 35 U.S.C. 112, sixth paragraph, must be identified and the structure, material, or acts described in the specification as corresponding to each claimed function must be set forth with reference to the specification by page and line number, and to the drawing, if any, by reference characters.” Again, the reference numbers and other bracketed information are illustrative only and are not intended to limit claims 27-30 and 49-51 to only the exact embodiments shown and described in the specification and figures of the present application.

27. An apparatus, comprising:

a network device [*108 in Figure 1 and Figures 3-4; page 7, lines 8-25*] adapted to be communicatively coupled between a client terminal [*102 in Figures 1-4; page 7, lines 1-3*] and a server [*114 in Figures 1-4; page 7, lines 21-25*], the network device having:

communication terminal means for establishing a client-side connection and a server-side connection [*Figures 1 and 3-4: terminal of the switch/gateway 108 coupled to the client 102 and terminal of the switch/gateway 108 coupled to the server 114; client-side connection line between client 102 and switch/gateway 108 in Figures 3-4; and server-side connection line 304/402 between switch/gateway 108 and server 114 in Figures 3-4; page 20, line 11; page 8, lines 4-17*]; and

means for reading content of a packet received via the client-side connection, and for extending persistency [110 and 112 in Figure 1; page 7, lines 11-15; page 5, lines 6-7; Request 1' in Figure 3; Request 1' at 404 in Figure 4; page 8, lines 18-28; page 9, lines 1-7; page 10, lines 3-16; page 11, lines 5-10; page 11, line 16 to page 12, line 3; and page 12, lines 4-14] of the server-side connection if the read content of the packet signals said server-side connection to close [Request 1 at 202 in Figure 2; Request 1 at 300 in Figure 3; Request 1 at 400 in Figure 4; page 9, lines 22-26; page 10, lines 3-5; and page 12, lines 5-9].

28. The apparatus of claim 27 wherein the means for extending the persistency of the server-side connection modifies header information in the packet to a format unrecognizable [*"Connection: Cselo"* in Request 1' in Figure 3; page 11, line 16 to page 12, line 3] to a server coupled to said server-side connection to receive said packet .

29. The apparatus of claim 27 wherein the means for extending the persistency of the server-side connection modifies header information in the packet to indicate a protocol version that corresponds to a persistent connection [*"HTTP/1.1"* in Request 1' in Figure 4; page 12, lines 4-14].

30. The apparatus of claim 27 wherein the means for extending the persistency of the server-side connection de-links the server-side connection from the client-side connection in response to RESET content in the packet received via the client side connection [page 8, lines 18-28].

49. The apparatus of claim 27 wherein the means for extending the persistency of the server-side connection, if the read content of the packet signals said server-side connection to close, maintains persistency by de-linking the server-side connection from the client-side connection in response to FIN content in the packet received via the client side connection [page 9, lines 1-11].

50. The apparatus of claim 27 wherein the means for extending the persistency of the server-side connection, if the read content of the packet signals said server-side connection to close, maintains persistency by de-linking the server-side connection from the client-side connection in response to RESET content in the packet received via the client side connection [*page 8, lines 18-28*].

51. The apparatus of claim 27 wherein the means for extending the persistency of the server-side connection, if the read content of the packet signals said server-side connection to close, maintains persistency by replacing a Connection: Close header of the packet with a Connection: Keep-Alive header [*Request 1 at 300 in Figure 3; page 10, line 3 to page 11, line 4*].

VI. GROUND OF REJECTION TO BE REVIEWED ON APPEAL

Whether claims 1-2, 6-15, 18-21, and 43-45 are unpatentable under 35 U.S.C. § 103(a) over Yang (U.S. Patent No. 7,216,172) in view of Erickson (U.S. Patent No. 6,721,792).

Whether claims 3-5 and 16-17 are unpatentable under 35 U.S.C. § 103(a) over Yang in view of Erickson, and further in view of Kirby (U.S. Patent No. 5,828,846).

Whether claims 22-36 and 46-52 are unpatentable “for the same reasons set forth to rejecting claims 1-21 and 43-45 above.” It is noted herein that page 7 (section 16) of the final Office Action of June 12, 2008 stated that claims 22-36 and 46-52 stand rejected “for the same reasons set forth to rejecting claims 1-21 and 43-45 above, since claims 22-36 and 46-52 are merely an apparatus for the method of operations defined in the method claims 1-21 and 43-45,” but did not further specifically enumerate or otherwise set forth in detail the statutory grounds for rejection and the references to support the rejections, on a claim-by-claim basis. Since page 7 (section 16) stated that claims 22-36 and 46-52 are unpatentable “for the same reasons set forth to rejecting claims 1-21 and 43-45 above” without giving further detail, it is assumed herein for purposes of identifying the grounds for rejection to be reviewed on appeal (and as can best be understood from the final Office Action of June 12, 2008) that claims 22, 24-29, 31-33, 35, 46, 48, and 51 are rejected under 35 U.S.C. § 103(a) as being unpatentable over Yang in view of Erickson, and that claims 23, 30, 34, 36, 47, 49-50, and 52 are rejected under 35 U.S.C. § 103(a) as being unpatentable over Yang in view of Erickson, and in further view of Kirby.

VII. ARGUMENT

A. Nonobviousness under Section 103

Consistent with a long line of judicial holdings, MPEP § 2143.03 states that “All words in a claim must be considered in judging the patentability of that claim against the prior art. *In re Wilson*, 424 F.2d 1382, 1385, 165 USPQ 494, 496 (CCPA 1970).” In order to find *prima facie* obviousness when combining references, MPEP § 2143(A)(1) states the following (emphasis ours): “Office personnel must articulate the following: (1) a finding that the prior art included each element claimed, although not necessarily in a single prior art reference, with the only difference between the claimed invention and the prior art being the lack of actual combination of the elements in a single prior art reference.” MPEP § 706.02(j) further states (emphasis ours): “To support the conclusion that the claimed invention is directed to obvious subject matter, either the references must expressly or impliedly suggest the claimed invention or the examiner must present a convincing line of reasoning as to why the artisan would have found the claimed invention to have been obvious in light of the teachings of the references. *Ex parte Clapp*, 227 USPQ 972, 973 (Bd. Pat. App. & Inter. 1985).”

The rejections of claims 1-36 and 43-52 do not meet these requirements for a rejection based on obviousness, since the cited references (whether singly or in combination) do not teach or suggest all of the claim limitations.

B. Rejection of independent claim 1 on the basis of Yang and Erickson

Independent claim 1 recites, *inter alia*, the following limitations (emphasis ours):

“receiving at said network device via said client-side connection a communication that signals said server-side connection to close; and
maintaining persistent, by said network device, at least the server-side connection in response to said communication received via said client-side connection.”

Page 3 (section 5) of the final Office Action of June 12, 2008 has in fact admitted that Yang does not teach “receiving at the network device via the client-side connection a communication that signals the server-side connection to close.”

Indeed, as explained on pages 13-14 of the amendment of March 5, 2008, Yang’s column 2, lines 21-41; column 5, lines 51-53; and column 8, lines 17-19 teach a technique to create a quasi-persistent connection “by modifying the content-length key-value pair in a HTTP request or response” so that “the content-length key-value pair has a content-length value that is large”—in this manner, the “client or server that receives the content-length key-value pair having the large content-length value, in effect, believes that a large amount of data will be included with the request or response,” and therefore does not close the connection.

It is therefore not possible for Yang to meet the limitations of claim 1 that require that the server-side connection is maintained persistent “in response to the communication received via said client-side connection” that “signals said server-side connection to close.” For instance, if Yang’s client modified the content-length key-value pair to extend persistency as taught by Yang, and then this modified content-length key-value pair is sent by Yang’s client to his server, the received modified content-length key-value pair would signal his server-side connection to remain open, rather than “signals said server-side connection to close” as required by claim 1.

To supply the missing teachings of Yang, page 3 (section 5) of the final Office Action of June 12, 2008 relies upon Erickson. However, Erickson does not cure the deficiencies of Yang.

Erickson’s invention is directed towards HTTP version 1.1 communications, as Erickson explicitly teaches below in his column 8, lines 19-34 (emphasis ours):

“FIGS. 5-7 illustrate formats for the communication flows of FIG. 4 in more detail. First, FIG. 5 illustrates a standard HTTP/1.1 post request format suitable for use in the present invention. The format of the post request is as specified in the HyperText Transfer Protocol HTTP/1.1 specification. The present invention is concerned with the contents of the post request rather than the format of the post request ... At line 202, an

HTTP-version field must specify version 1.1 or greater. Only version 1.1 or greater allows a transfer-encoding to be chunked as is necessary for the present embodiment of the invention. On line 204, the transfer-encoding is designated as chunked.”

As abundantly clear from the above-quoted passage from Erickson, his invention is directed towards an HTTP 1.1 implementation. As described in pages 1-4 of the present application, HTTP 1.1 provides persistent connections as a default, and Erickson’s column 2, lines 29-30 also concurs that HTTP 1.1 provides a persistent connection (“the newer HTTP 1.1 specification provides a keep-alive mechanism that allows one connection for multiple objects on an HTML page”). Accordingly, the HTTP 1.1 implementation of Erickson’s invention does not in and of itself address “a communication that signals said server-side connection to close” of claim 1.

Rather, Erickson’s invention provides a persistency technique to addresses the situation in an HTTP 1.1 implementation where a connection might potentially be closed “after a period of inactivity.” Erickson teaches the following in his column 2, lines 27-32 and column 9, lines 1-32 (emphasis ours):

“Even though the newer HTTP 1.1 specification provides a keep-alive mechanism that allows one connection for multiple objects on an HTML page, the connection is closed either by the Web server or the browser after a period of inactivity ... FIG. 7 illustrates a format for a chunk suitable for use in the present invention. At line 252, the first field specifies an ASCII representation of the hex length of the chunk. According to the HTTP 1.1 specification, by allowing each chunk to specify the hex length of the chunk, multiple chunks (messages) may be sent for one HTTP request. However, the last message must provide a unique identifier to indicate when the HTTP request is complete. Typically, the Web server software waits until all the chunks for one message body are received before generating a response. However,

because the present invention uses the chunking option in a new manner and does not send a last message, the present invention prevents the Web server software from buffering the chunked messages and instead forces the Web server software to send the chunked messages to the extension as the chunked messages are received. This allows the present invention to provide bi-directional communication without waiting for an entire message and without establishing more than one connection. At line 254, an end of chunk header indicator is chosen according to the HTTP/1.1 specification. Lines 256 and 258 provide the tunneling aspect of the present invention. First, line 256 provides a data field that may be sent in all chunked packets to keep the connection alive. This Keep-Alive field is necessary because some Web servers will close a connection if additional chunks are not received within a certain time frame. Therefore, in the present invention, the HTTP tunnel mechanism that is running on the Web client generates a chunk with only the Keep-Alive data field and without any additional Telnet data after a predetermined period of inactivity. The sending of only the Keep-Alive data in a chunk allows the connection to remain alive even during periods of inactivity. Line 258 represents the tunneling of the Telnet messages generated by either the host system or the Web client emulator. The Telnet messages generated by either the host system or the Web client emulator are written into the Telnet message field. Line 260 specifies the end of chunk indicator as specified in the HTTP/1.1 specification.”

From the above-quoted passages of Erickson, it is again abundantly clear that his invention is directed towards an HTTP 1.1 implementation, in particular use of the “chunking” aspect of HTTP 1.1. As taught by Erickson above, his invention “generates a chunk with only the Keep-Alive data field” and then sends this chunk having the Keep-Alive data, which “allows the connection to remain alive even during periods of inactivity.”

At least two things can be readily gleaned from Erickson's teachings above. First, his chunk having the "Keep-Alive" data can in no way meet the limitations of claim 1 that require "a communication that signals said server-side connection to close." The communication of claim 1 "signals the server-side connection to close," whereas Erickson's chunk (having the "Keep Alive" data) signals his "connection to remain alive." These are opposite/contradictory teachings.

Second, Erickson's teachings above address the situation where a connection might close "if additional chunks are not received within a certain time frame" and "during periods of inactivity." It is implicit/explicit therefore, that there is no communications of any sort (in particular "a communication that signals said server-side connection to close") that is being sent/received during these inactivity periods or time frame of Erickson. Thus, Erickson again does not meet the limitations of claim 1 that require "receiving ... a communication that signals said server-side connection to close" and "maintaining persistent ... at least the server-side connection in response to said communication," since Erickson is addressing a situation where there is no communication ("periods of inactivity") at all and therefore has to provide the chunk having the Keep-Alive data in order to prevent the connection from closing due to inactivity.

Page 3 (section 5) of the final Office Action cites Erickson's column 2, lines 22-39 as allegedly disclosing "a communication that signals the server-side connection to close." However, a thorough and accurate reading of this passage reveals that Erickson is merely discussing (in column 2, lines 22-27) the HTTP versions prior to HTTP 1.1. In these earlier versions prior to HTTP 1.1 (and as also explained in pages 1-4 of the present application), the connection is closed after the response is sent. This cited passage of Erickson, while speaking of closing of a connection after sending a response, is not applicable to and has nothing to do with his invention. As previously explained above, Erickson's invention is explicitly directed towards (and limited to) HTTP 1.1 that uses chunking and wherein the default behavior is to keep a connection persistent, except after a period of inactivity—Erickson teaches an invention to maintain the connection persistent to prevent closure due to a period of inactivity in an HTTP 1.1 implementation—Erickson is completely silent as to any applicability and as to any possible manner of implementation of his invention with respect to other HTTP versions (per his column 2, lines 22-27) that existed prior to HTTP 1.1 and that closed a connection after sending a

response. Thus, the passage in column 2, lines 22-27 of Erickson discussing HTTP versions that were prior to HTTP 1.1, considered as a whole along with his other teachings, does not meet the limitations of claim 1 that require “receiving a communication that signals said server-side connection to close” and “maintaining persistent at least the server-side connection in response to said communication.”

In view of the above, since all of the limitations of claim 1 are not met by the cited references, claim 1 is not obvious.

C. Rejection of dependent claims 2, 8, 13, and 43 on the basis of Yang and Erickson

Dependent claims 2, 8, 13, and 43 depend on claim 1, and by virtue of this dependency, are nonobvious over the cited references for the reasons set forth above with respect to claim 1.

D. Rejection of dependent claim 5 on the basis of Yang, Erickson, and Kirby

Dependent claim 5 also depends on claim 1, and by virtue of this dependency, is also nonobvious over the cited references of Yang and Erickson for the reasons set forth above with respect to claim 1. In rejecting claim 5, page 6 (section 13) of the final Office Action of June 12, 2008 also relies upon Kirby. However, Kirby does not teach and has not been cited as teaching at least the limitations of claim 5 (from its base claim 1) involving “receiving a communication that signals said server-side connection to close” and “maintaining persistent at least the server-side connection in response to said communication.” Hence, claim 5 is not obvious.

E. Rejection of dependent claim 3 on the basis of Yang, Erickson, and Kirby

Dependent claim 3 depends on claim 1, and by virtue of this dependency, is nonobvious over the cited references for the reasons set forth above with respect to claim 1.

Dependent claim 3 is also nonobvious due to the limitations recited therein. In particular, dependent claim 3 recites, *inter alia*, the following limitations (emphasis ours):

“wherein maintaining persistent at least the server-side connection
in response to said communication received via said client-side connection
includes:

de-linking, by said network device, the server-side connection
from the client-side connection in response to a RESET packet received
by said network device via said client-side connection.”

These limitations are not met by the cited references. Page 6 (section 14) of the final Office Action of June 12, 2008 has admitted that neither Yang nor Erickson teach “de-linking, by said network device, the server-side connection from the client-side connection in response to a RESET packet.” To supply the missing teachings of Yang and Erickson, the final Office Action of June 12, 2008 relies upon Kirby.

However, Kirby does not cure the deficiencies of Yang and Erickson.

Specifically, claim 3 requires “maintaining persistent at least the server-side connection ... in response to a RESET packet.” Kirby does not teach maintaining a server-side connection persistent in response to a RESET packet. Quite the opposite to claim 3, Kirby teaches terminating a connection in response to a RESET (RST) packet—Kirby is completely silent as to any relationship between his RST packet and maintaining a persistent connection as recited in claim 3.

For example, Kirby teaches the following in his column 4, lines 8-11 (emphasis ours):

“The connection packet handler 108 watches 120 (FIG. 7) packets passing
on link 20 for those of types involved in setting up and terminating
connections (e.g., SYN, FIN, and RST packets).”

From the above-quoted passage of Kirby, he teaches that his RST packet is “involved in setting up and terminating connections.” Accordingly, Kirby does not meet the limitations of claim 3 that require “maintaining persistent at least the server-side connection ... in response to a RESET packet.”

In view of the above, claim 3 is not obvious.

F. Rejection of dependent claim 4 on the basis of Yang, Erickson, and Kirby

Dependent claim 4 depends on claim 1, and by virtue of this dependency, is nonobvious over the cited references for the reasons set forth above with respect to claim 1. Dependent claim 4 is also nonobvious due to the limitations recited therein. In particular, dependent claim 4 recites, *inter alia*, the following limitations (emphasis ours):

“wherein maintaining persistent at least the server-side connection
in response to said communication received via said client-side connection
includes:

de-linking, by said network device, the server-side connection
from the client-side connection in response to a FIN packet received by
said network device via said client-side connection.”

These limitations are not met by the cited references. Page 6 (section 14) of the final Office Action of June 12, 2008 has admitted that neither Yang nor Erickson teach “de-linking, by said network device, the server-side connection from the client-side connection ... in response to a FIN packet.” To supply the missing teachings of Yang and Erickson, the final Office Action of June 12, 2008 relies upon Kirby.

However, Kirby does not cure the deficiencies of Yang and Erickson.

Specifically, claim 4 requires “maintaining persistent at least the server-side connection ... in response to a FIN packet.” Kirby does not teach maintaining a server-side connection persistent in response to a FIN packet. Quite the opposite to claim 4, Kirby teaches

terminating a connection in response to a FIN packet—Kirby is completely silent as to any relationship between his RST packet and maintaining a persistent connection as recited in claim 4.

For example, Kirby teaches the following in his column 4, lines 59-67 (emphasis ours):

“The packet sorter sends FIN packets to a FIN packet processor 154. The FIN packet processor sends the packets to the state processor 146. The state processor determines 184 if the FIN packet is a response to a previously allowed FIN packet (indicating that device 12 and device 26 have agreed to terminate their communications session) by looking in the log table 148 for a FIN packet from device 26. If so, the state processor modifies 186 the state table to delete the connection between device 12 and device 26.”

From the above-quoted passage of Kirby, it is abundantly clear that his FIN packet is used to “delete the connection between device 12 and device 26.” Accordingly, Kirby does not meet the limitations of claim 4 that require “maintaining persistent at least the server-side connection ... in response to a FIN packet.”

In view of the above, claim 4 is not obvious.

G. Rejection of dependent claims 6-7 on the basis of Yang and Erickson

Dependent claim 6 depends on claim 1, and by virtue of this dependency, is nonobvious over the cited references for the reasons set forth above with respect to claim 1. Dependent claim 6 is also nonobvious due to the limitations recited therein. In particular, dependent claim 6 recites, *inter alia*, the following limitations (emphasis ours):

“wherein maintaining persistent at least the server-side connection
in response to said communication received via said client-side connection
includes:

identifying, by said network device, a Connection: Close header in the communication received via the client-side connection; and

replacing, by said network device, the Connection: Close header in the communication with a Connection: Keep-Alive header.”

These limitations are not met by the cited references. The final Office Action of June 12, 2008 has not cited any teaching of Yang that meets these limitations of claim 6. Instead, page 4 (section 6) of the final Office Action of June 12, 2008 relies upon column 7, line 16 to column 8, line 67 of Erickson, a portion of which includes column 8, lines 19-54 reproduced below (emphasis ours):

“FIGS. 5-7 illustrate formats for the communication flows of FIG. 4 in more detail. First, FIG. 5 illustrates a standard HTTP/1.1 post request format suitable for use in the present invention. The format of the post request is as specified in the HyperText Transfer Protocol HTTP/1.1 specification. The present invention is concerned with the contents of the post request rather than the format of the post request ... At line 202, an HTTP-version field must specify version 1.1 or greater. Only version 1.1 or greater allows a transfer-encoding to be chunked as is necessary for the present embodiment of the invention ... At line 212, in the Connection field, Keep-Alive is specified so that the connection remains active. However, even with Keep-Alive specified, the connection may be terminated by either the Web client or the Web server after even very brief periods of idleness. In order to achieve a true Keep-Alive connection, the present invention incorporates additional Keep-Alive data in the tunneled data as will be described in greater detail with reference to FIG. 7.”

Thus from the above-quoted passage of Erickson relied upon by the final Office Action of June 12, 2008 to reject claim 6, it is abundantly clear that Erickson is silent as to any identifying or replacing of a “Connection: Close header” as recited in claim 6. Instead, Erickson

merely teaches using Keep-Alive data in order to keep his connection alive, where the connection might otherwise be closed due to “periods of idleness.” “Periods of idleness” where no communications are received are not the same as the claimed receiving of a communication that has a Connection: Close header of claim 6.

In view of the above, claim 6 is not obvious.

Dependent claim 7 depends on claim 6, and by virtue of this dependency, is nonobvious over the cited references for the reasons set forth above with respect to claim 6.

H. Rejection of dependent claims 9-10 on the basis of Yang and Erickson

Dependent claim 9 depends on claim 1, and by virtue of this dependency, is nonobvious over the cited references for the reasons set forth above with respect to claim 1. Dependent claim 9 is also nonobvious due to the limitations recited therein. In particular, dependent claim 9 recites, *inter alia*, the following limitations (emphasis ours):

“wherein maintaining persistent at least the server-side connection in response to said communication received via said client-side connection includes:

modifying, by said network device, a header in the communication from a format that signals the server-side connection to close to a format that is unrecognizable by a server coupled to said server-side connection, to cause the server to ignore the modified header.”

These limitations are not met by the cited references. The final Office Action of June 12, 2008 has not cited any teaching of Yang that meets these limitations of claim 9. Instead, page 4 (section 7) of the final Office Action of June 12, 2008 relies upon column 6, line 28 to column 8, line 67 of Erickson, a portion of which includes column 6, line 58 to column 7, line 16 reproduced below (emphasis ours):

“Depending upon the type of Web server 120, a setting may need to be changed to inform the Web server 120 not to process chunked data

designating the extension 132. By changing the setting, the extension is allowed to handle the chunked data in a manner described in detail below. To change the setting of any Web server 120 running Microsoft Internet Information Server (IIS) 4.0 by Microsoft Corporation, the extension 132 accesses a metabase and programmatically changes an MD_UPLOAD_READAHEAD_SIZE parameter to zero (0) for the named extension 132. The MD_UPLOAD_READAHEAD_SIZE parameter is typically used to fine tune memory usage on the Web server 120. Typically, the MD_UPLOAD_READAHEAD_SIZE parameter is set to a value that is an average size of a request message received from a Web client 126. The present invention therefore modifies the MD_UPLOAD_READAHEAD_SIZE parameter for an unusual purpose of having the Web server software 121 ignore processing of chunked data that is designated to be passed to the extension 132. By disabling the processing of chunked data in the Web server software 121, the extension 132 can provide the processing for the chunked data, thereby allowing a connection-oriented protocol to be tunneled on top of HTTP and creating a bi-directional connection that remains active for the duration of the communication between a Web client and a host system.”

Thus from the above-quoted passage of Erickson relied upon by the final Office Action of June 12, 2008 to reject claim 9, it is abundantly clear that Erickson is silent as to any header having “a format that signals the server-side connection to close” as recited in claim 9. Instead, Erickson teaches the “MD_UPLOAD_READAHEAD_SIZE parameter” that is typically “set to a value that is an average size of a request message received,” and then modifying this parameter in order to have the web server “ignore processing of chunked data that is designated to be passed to the extension 132.” The “MD_UPLOAD_READAHEAD_SIZE parameter” of Erickson is not the same as the claimed header having “a format that signals the server-side connection to close,” since the MD_UPLOAD_READAHEAD_SIZE parameter has to provide “a value that is an average size of a request message received,” thereby implicitly/explicitly

teaching an open connection to receive/process the request message, rather than a “a format that signals the server-side connection to close” as required in claim 9.

Furthermore, Erickson’s teaches in column 6, line 58 to column 7, line 16 quoted above that modification of the “MD_UPLOAD_READAHEAD_SIZE parameter” involves modification of a parameter in the web server 120 by the extension 132. In contrast, claim 9 requires modification of “a header in the communication.” Modification of “a header in the communication” is not the same as Erickson’s modification of a parameter in the web server 120.

In view of the above, claim 9 is not obvious.

Dependent claim 10 depends on claim 9, and by virtue of this dependency, is nonobvious over the cited references for the reasons set forth above with respect to claim 9.

I. Rejection of dependent claims 11-12 and 44 on the basis of Yang and Erickson

Dependent claim 11 depends on claim 1, and by virtue of this dependency, is nonobvious over the cited references for the reasons set forth above with respect to claim 1. Dependent claim 11 is also nonobvious due to the limitations recited therein. In particular, dependent claim 11 recites, *inter alia*, the following limitations (emphasis ours):

“wherein maintaining persistent at least the server-side connection
in response to said communication received via said client-side connection
includes:

changing, by said network device, a HTTP version value indicated
in the communication to another HTTP version value that is recognizable
by a server, coupled to said server-side connection, as being associated
with a persistent connection.”

These limitations are not met by the cited references. The final Office Action of June 12, 2008 has not cited any teaching of Yang that meets these limitations of claim 11. Instead, page 5 (section 8) of the final Office Action of June 12, 2008 relies upon column 8, line

19 to column 9, line 38 of Erickson, a portion of which includes column 8, lines 55-58 reproduced below (emphasis ours):

“FIG. 6 illustrates a format for a standard HTTP/1.1 response. Again, the format of the data is as specified in the HTTP/1.1 specification and the present invention is only concerned with the content of the fields.”

Thus from the above-quoted passage of Erickson relied upon by the final Office Action of June 12, 2008 to reject claim 11, it is abundantly clear that Erickson is silent as to any changing of an HTTP version value as recited in claim 11. Instead, Erickson teaches that his “invention is only concerned with the content of the fields” of the HTTP/1.1 specification. Erickson provides absolutely no teaching with respect to changing an HTTP version value (*e.g.*, an HTTP version other than HTTP 1.1.) indicated in his fields to another HTTP version value, such as HTTP 1.1.

In view of the above, claim 11 is not obvious.

Dependent claims 12 and 44 depend on claim 11, and by virtue of this dependency, are nonobvious over the cited references for the reasons set forth above with respect to claim 11.

J. Rejection of independent claim 14 on the basis of Yang and Erickson

Independent claim 14 recites, *inter alia*, the following limitations (emphasis ours):

“extending, by said network device, persistence of the server-side connection if the read content of the packet signals said server-side connection to close.”

Page 3 (section 5) of the final Office Action of June 12, 2008 has admitted that Yang does not teach “receiving at the network device via the client-side connection a communication that signals the server-side connection to close.”

As explained on pages 13-14 of the amendment of March 5, 2008, Yang’s column 2, lines 21-41; column 5, lines 51-53; and column 8, lines 17-19 teach a technique to create a

quasi-persistent connection “by modifying the content-length key-value pair in a HTTP request or response” so that “the content-length key-value pair has a content-length value that is large”—in this manner, the “client or server that receives the content-length key-value pair having the large content-length value, in effect, believes that a large amount of data will be included with the request or response,” and therefore does not close the connection.

It is therefore not possible for Yang to meet the limitations of claim 14 that require “extending ... persistency of the server-side connection if the read content ... signals said server-side connection to close.” For instance, if Yang’s client modified the content-length key-value pair to extend persistency as taught by Yang, and then this modified content-length key-value pair is sent by Yang’s client to his server, the received modified content-length key-value pair would signal his server-side connection to remain open, rather than “signals said server-side connection to close” as required by claim 14.

To supply the missing teachings of Yang, page 3 (section 5) of the final Office Action of June 12, 2008 relies upon Erickson. However, Erickson does not cure the deficiencies of Yang.

Erickson’s invention is directed towards HTTP version 1.1 communications, as Erickson explicitly teaches in his column 8, lines 19-34 (previously quoted above). As abundantly clear from the above-quoted passage of column 8, lines 19-34 from Erickson, his invention is directed towards and is limited to an HTTP 1.1 implementation. As described in pages 1-4 of the present application, HTTP 1.1 provides persistent connections as a default, and Erickson’s column 2, lines 29-30 also concurs that HTTP 1.1 provides a persistent connection (“the newer HTTP 1.1 specification provides a keep-alive mechanism that allows one connection for multiple objects on an HTML page”). Accordingly, the HTTP 1.1 implementation of Erickson’s invention does not in and of itself address “content of the packet [that] signals said server-side connection to close” of claim 14.

Rather, Erickson’s invention provides a persistency technique to addresses the situation in an HTTP 1.1 implementation where a connection might potentially be closed “after a period of inactivity.” In his column 2, lines 27-32 and column 9, lines 1-32 (previously quoted above), Erickson teaches rather clearly that his invention is directed towards an HTTP 1.1 implementation, in particular use of the “chunking” aspect of HTTP 1.1. As taught by Erickson,

his invention “generates a chunk with only the Keep-Alive data field” and then sends this chunk having the Keep-Alive data, which “allows the connection to remain alive even during periods of inactivity.”

At least two things can be readily gleaned from Erickson’s teachings in his column 2, lines 27-32 and column 9, lines 1-32. First, his chunk having the “Keep-Alive” data can in no way meet the limitations of claim 14 that require “content of a packet [that] signals said server-side connection to close.” The packet content of claim 14 “signals the server-side connection to close,” whereas Erickson’s chunk (having the “Keep Alive” data) signals his “connection to remain alive.” These are opposite/contradictory teachings.

Second, Erickson’s invention addresses the situation where a connection might close “if additional chunks are not received within a certain time frame” and “during periods of inactivity.” It is implicit/explicit therefore, that there is no communications of any sort (in particular “content of the packet [that] signals said server-side connection to close”) that is being sent/received during these inactivity periods or time frame of Erickson. Thus, Erickson again does not meet the limitations of claim 14 that require “extending ... persistency ... if the read content of the packet signals said server-side connection to close,” since Erickson is addressing a situation where there is no communication (“periods of inactivity”) at all and therefore has to provide the chunk having the Keep-Alive data in order to prevent the connection from closing due to inactivity.

Page 3 (section 5) of the final Office Action cites Erickson’s column 2, lines 22-39 as allegedly disclosing “a communication that signals the server-side connection to close.” However, a thorough and accurate reading of this passage reveals that Erickson is merely discussing (in column 2, lines 22-27) the HTTP versions prior to HTTP 1.1. In these earlier versions prior to HTTP 1.1 (and as also explained in pages 1-4 of the present application), the connection is closed after the response is sent. This cited passage of Erickson, while speaking of closing of a connection after sending a response, is not applicable to and has nothing to do with his invention. As previously explained above, Erickson’s invention is explicitly directed towards (and limited to) HTTP 1.1 that uses chunking and wherein the default behavior is to keep a connection persistent, except after a period of inactivity—Erickson teaches an invention to maintain the connection persistent to prevent closure due to a period of inactivity in an HTTP 1.1.

implementation—Erickson is completely silent as to any applicability and as to any possible manner of implementation of his invention with respect to other HTTP versions (per his column 2, lines 22-27) that existed prior to HTTP 1.1 and that closed a connection after sending a response. Thus, the passage in column 2, lines 22-27 of Erickson discussing HTTP versions that were prior to HTTP 1.1, considered as a whole along with his other teachings, does not meet the limitations of claim 14 that require “extending ... persistency ... if the read content of the packet signals said server-side connection to close.”

In view of the above, since all of the limitations of claim 14 are not met by the cited references, claim 14 is not obvious.

K. Rejection of dependent claims 15, 19, and 45 on the basis of Yang and Erickson

Dependent claims 15, 19, and 45 depend on claim 14, and by virtue of this dependency, are nonobvious over the cited references for the reasons set forth above with respect to claim 14.

L. Rejection of Dependent claim 16 on the basis of Yang, Erickson, and Kirby

Dependent claim 16 depends on claim 14, and by virtue of this dependency, is nonobvious over the cited references for the reasons set forth above with respect to claim 14. Dependent claim 16 is also nonobvious due to the limitations recited therein. In particular, dependent claim 16 recites, *inter alia*, the following limitations (emphasis ours):

“wherein extending, by said network device, the persistency of the server-side connection includes:

de-linking, by said network device, the server-side connection from the client-side connection, if the read content indicates that the packet is a RESET packet received via the client-side connection.”

These limitations are not met by the cited references. Page 7 (section 15) of the final Office Action of June 12, 2008 has admitted that neither Yang nor Erickson teach de-linking the server-side connection from the client-side connection, if the read content indicates that the packet is a RESET packet. To supply the missing teachings of Yang and Erickson, the final Office Action of June 12, 2008 relies upon Kirby.

However, Kirby does not cure the deficiencies of Yang and Erickson.

Specifically, claim 16 requires “extending ... the persistence of the server-side connection ... if the read content indicates that the packet is a RESET packet.” Kirby does not teach extending persistence if a received packet is a RESET packet. Quite the opposite to claim 16, Kirby teaches terminating a connection in response to a RESET (RST) packet—Kirby is completely silent as to any relationship between his RST packet and extending persistence as recited in claim 16.

For example, Kirby teaches the following in his column 4, lines 8-11 (emphasis ours):

“The connection packet handler 108 watches 120 (FIG. 7) packets passing on link 20 for those of types involved in setting up and terminating connections (e.g., SYN, FIN, and RST packets).”

From the above-quoted passage of Kirby, he teaches that his RST packet is “involved in setting up and terminating connections.” Accordingly, Kirby does not meet the limitations of claim 16 that require “extending ... the persistence ... if the read content indicates that the packet is a RESET packet.”

In view of the above, claim 16 is not obvious.

M. Rejection of dependent claim 17 on the basis of Yang, Erickson, and Kirby

Dependent claim 17 depends on claim 14, and by virtue of this dependency, is nonobvious over the cited references for the reasons set forth above with respect to claim 14.

Dependent claim 17 is also nonobvious due to the limitations recited therein. In particular, dependent claim 17 recites, *inter alia*, the following limitations (emphasis ours):

“wherein extending, by said network device, the persistency of the server-side connection includes:

de-linking, by said network device, the server-side connection from the client-side connection, if the read content indicates that the packet is a FIN packet received via the client-side connection.”

These limitations are not met by the cited references. Page 7 (section 15) of the final Office Action of June 12, 2008 has admitted that neither Yang nor Erickson teach de-linking the server-side connection from the client-side connection, if the read content indicates that the packet is a FIN packet. To supply the missing teachings of Yang and Erickson, the final Office Action of June 12, 2008 relies upon Kirby.

However, Kirby does not cure the deficiencies of Yang and Erickson.

Specifically, claim 17 requires that “extending ... the persistency of the server-side connection ... if the read content indicates that the packet is a FIN packet.” Kirby does not teach extending persistency if a received packet is a FIN packet. Quite the opposite to claim 17, Kirby teaches terminating a connection in response to a FIN packet—Kirby is completely silent as to any relationship between his FIN packet and extending persistency as recited in claim 17.

For example, Kirbys’ column 4, lines 59-67 (previously quoted above) teaches that his FIN packet is used “to delete the connection between device 12 and device 26.” Accordingly, Kirby does not meet the limitations of claim 17 that require “extending ... the persistency ... if the read content indicates that the packet is a FIN packet.”

In view of the above, claim 17 is not obvious.

N. Rejection of dependent claim 18 on the basis of Yang and Erickson

Dependent claim 18 depends on claim 14, and by virtue of this dependency, is nonobvious over the cited references for the reasons set forth above with respect to claim 14.

Dependent claim 18 is also nonobvious due to the limitations recited therein. In particular, dependent claim 18 recites, *inter alia*, the following limitations (emphasis ours):

“wherein extending, by said network device, the persistency of the server-side connection includes:

identifying, by said network device, header information of the packet received via the client-side connection that signals the server-side connection to close; and

replacing, by said network device, the identified header information with new header information that maintains the server-side connection persistent.”

These limitations are not met by the cited references. The final Office Action of June 12, 2008 has not cited any teaching of Yang that meets these limitations of claim 18. Instead, the final Office Action of June 12, 2008 relies upon column 7, line 16 to column 8, line 67 of Erickson, a portion of which includes column 8, lines 19-54 that was previously quoted above.

Erickson is silent as to any identifying or replacing of “header information ... that signals the server-side connection to close” as recited in claim 18. Instead, column 8, lines 19-54 of Erickson merely teaches using Keep-Alive data in order to keep his connection alive, where the connection might otherwise be closed due to “periods of idleness.” “Periods of idleness” where no communications are received are not the same as the claimed received packet having “header information ... that signals the server-side connection to close” of claim 18.

In view of the above, claim 18 is not obvious.

O. Rejection of dependent claim 20 on the basis of Yang and Erickson

Dependent claim 20 depends on claim 14, and by virtue of this dependency, is nonobvious over the cited references for the reasons set forth above with respect to claim 14. Dependent claim 20 is also nonobvious due to the limitations recited therein. In particular, dependent claim 20 recites, *inter alia*, the following limitations (emphasis ours):

“wherein extending, by said network device, the persistency of the server-side connection includes:

modifying, by said network device, header information in the packet received via the client-side connection from a format that signals said server-side connection to close to a format that is unrecognizable by a server that is to receive the packet via the server-side connection.”

These limitations are not met by the cited references. The final Office Action of June 12, 2008 has not cited any teaching of Yang that meets these limitations of claim 20. Instead, the final Office Action of June 12, 2008 relies upon column 6, line 28 to column 8, line 67 of Erickson, a portion of which includes column 6, line 58 to column 7, line 16 previously quoted above. In the passage of Erickson relied upon by the final Office Action of June 12, 2008, Erickson is silent as to any header information having “a format that signals the server-side connection to close” as recited in claim 20. Instead, Erickson teaches in column 6, line 58 to column 7, line 16 that the “MD_UPLOAD_READAHEAD_SIZE parameter” that is typically “set to a value that is an average size of a request message received,” and then modifying this parameter in order to have the web server “ignore processing of chunked data that is designated to be passed to the extension 132.” The “MD_UPLOAD_READAHEAD_SIZE parameter” of Erickson is not the same as the claimed header information having “a format that signals the server-side connection to close,” since the MD_UPLOAD_READAHEAD_SIZE parameter has to provide “a value that is an average size of a request message received,” thereby implicitly/explicitly teaching an open connection to receive/process the request message, rather than a “a format that signals the server-side connection to close” as required in claim 20.

Furthermore, Erickson’s teaches in column 6, line 58 to column 7, line 16 (quoted previously above) that modification of the “MD_UPLOAD_READAHEAD_SIZE parameter” involves modification of a parameter in the web server 120 by the extension 132. In contrast, claim 20 requires modification of “header information in the packet.” Modification of “header information in the packet” is not the same as Erickson’s modification of a parameter in the web server 120.

In view of the above, claim 20 is not obvious.

P. Rejection of dependent claim 21 on the basis of Yang and Erickson

Dependent claim 21 depends on claim 14, and by virtue of this dependency, is nonobvious over the cited references for the reasons set forth above with respect to claim 14. Dependent claim 21 is also nonobvious due to the limitations recited therein. In particular, dependent claim 21 recites, *inter alia*, the following limitations (emphasis ours):

“wherein extending, by said network device, the persistency of the server-side connection includes:

changing, by said network device, a protocol version value indicated in the packet received via the client-side connection to a different protocol version value that corresponds to maintaining a persistent connection.”

These limitations are not met by the cited references. The final Office Action of June 12, 2008 has not cited any teaching of Yang that meets these limitations of claim 21. Instead, the final Office Action of June 12, 2008 relies upon column 8, line 19 to column 9, line 38 of Erickson, a portion of which includes column 8, lines 55-58 previously quoted above. This passage of Erickson relied upon by the final Office Action of June 12, 2008 is silent as to any changing of a protocol version value as recited in claim 21. Instead, Erickson teaches in his column 8, lines 55-58 that his “invention is only concerned with the content of the fields” of the HTTP/1.1 specification. Erickson provides absolutely no teaching with respect to changing a protocol version value (*e.g.*, an HTTP version other than HTTP 1.1.) indicated in his fields to another HTTP version value, such as HTTP 1.1.

In view of the above, claim 21 is not obvious.

Q. Rejection of independent claim 22 on the basis of Yang and Erickson

Independent claim 22 recites, *inter alia*, the following limitations (emphasis ours):

“receive via said client-side connection a communication that signals said server-side connection to close; and

maintain persistent, by said network device, at least the server-side connection in response to said communication received via said client-side connection.”

Page 3 (section 5) of the final Office Action of June 12, 2008 has admitted that Yang does not teach “receiving at the network device via the client-side connection a communication that signals the server-side connection to close.”

As explained on pages 13-14 of the amendment of March 5, 2008, Yang’s column 2, lines 21-41; column 5, lines 51-53; and column 8, lines 17-19 teach a technique to create a quasi-persistent connection “by modifying the content-length key-value pair in a HTTP request or response” so that “the content-length key-value pair has a content-length value that is large”—in this manner, the “client or server that receives the content-length key-value pair having the large content-length value, in effect, believes that a large amount of data will be included with the request or response,” and therefore does not close the connection.

It is therefore not possible for Yang to meet the limitations of claim 22 that require that the server-side connection is maintained persistent “in response to said communication received via said client-side connection” that “signals said server-side connection to close.” For instance, if Yang’s client modified the content-length key-value pair to provide persistency as taught by Yang, and then this modified content-length key-value pair is sent by Yang’s client to his server, the received modified content-length key-value pair would signal his server-side connection to remain open, rather than “signals said server-side connection to close” as required by claim 22.

To supply the missing teachings of Yang, the final Office Action of June 12, 2008 relies upon Erickson. However, Erickson does not cure the deficiencies of Yang.

Erickson’s invention is directed towards a HTTP version 1.1 implementation, as Erickson explicitly teaches in his column 8, lines 19-34 (previously quoted above). As described in pages 1-4 of the present application, HTTP 1.1 provides persistent connections as a default, and Erickson’s column 2, lines 29-30 also concurs that HTTP 1.1 provides a persistent connection (“the newer HTTP 1.1 specification provides a keep-alive mechanism that allows one connection for multiple objects on an HTML page”). Accordingly, the HTTP 1.1 implementation of

Erickson's invention does not in and of itself address "a communication that signals said server-side connection to close" of claim 22.

Rather, Erickson's invention provides a persistency technique to addresses the situation in an HTTP 1.1 implementation where a connection might potentially be closed "after a period of inactivity." In his column 2, lines 27-32 and column 9, lines 1-32 (previously quoted above), Erickson teaches rather clearly that his invention is directed towards an HTTP 1.1 implementation, in particular use of the "chunking" aspect of HTTP 1.1. As taught by Erickson, his invention "generates a chunk with only the Keep-Alive data field" and then sends this chunk having the Keep-Alive data, which "allows the connection to remain alive even during periods of inactivity."

At least two things can be readily gleaned from Erickson's teachings in his column 2, lines 27-32 and column 9, lines 1-32. First, his chunk having the "Keep-Alive" data can in no way meet the limitations of claim 22 that require "a communication that signals said server-side connection to close." The communication of claim 22 "signals the server-side connection to close," whereas Erickson's chunk (having the "Keep Alive" data) signals his "connection to remain alive." These are opposite/contradictory teachings.

Second, Erickson's invention addresses the situation where a connection might close "if additional chunks are not received within a certain time frame" and "during periods of inactivity." It is implicit/explicit therefore, that there is no communications of any sort (in particular "a communication that signals said server-side connection to close") that is being sent/received during these inactivity periods or time frame of Erickson. Thus, Erickson again does not meet the limitations of claim 22 that require "receive ... a communication that signals said server-side connection to close" and "maintain persistent ... at least the server-side connection in response to said communication," since Erickson is addressing a situation where there is no communication ("periods of inactivity") at all and therefore has to provide the chunk having the Keep-Alive data in order to prevent the connection from closing due to inactivity.

Page 3 (section 5) of the final Office Action cites Erickson's column 2, lines 22-39 as allegedly disclosing "a communication that signals the server-side connection to close." However, a thorough and accurate reading of this passage reveals that Erickson is merely discussing (in column 2, lines 22-27) the HTTP versions prior to HTTP 1.1. In these earlier

versions prior to HTTP 1.1 (and as also explained in pages 1-4 of the present application), the connection is closed after the response is sent. This cited passage of Erickson, while speaking of closing of a connection after sending a response, is not applicable to and has nothing to do with his invention. As previously explained above, Erickson's invention is explicitly directed towards (and limited to) HTTP 1.1 that uses chunking and wherein the default behavior is to keep a connection persistent, except after a period of inactivity—Erickson teaches an invention to maintain the connection persistent to prevent closure due to a period of inactivity in an HTTP 1.1 implementation—Erickson is completely silent as to any applicability and as to any possible manner of implementation of his invention with respect to other HTTP versions (per his column 2, lines 22-27) that existed prior to HTTP 1.1 and that closed a connection after sending a response. Thus, the passage in column 2, lines 22-27 of Erickson discussing HTTP versions that were prior to HTTP 1.1, considered as a whole along with his other teachings, does not meet the limitations of claim 22 that require “receive ... a communication that signals said server-side connection to close” and “maintain persistent ... at least the server-side connection in response to said communication.”

In view of the above, since all of the limitations of claim 22 are not met by the cited references, claim 22 is not obvious.

R. Rejection of dependent claims 24 and 46 on the basis of Yang and Erickson

Dependent claims 24 and 46 depend on claim 22, and by virtue of this dependency, are nonobvious over the cited references for the reasons set forth above with respect to claim 22.

S. Rejection of dependent claim 23 on the basis of Yang, Erickson, and Kirby

Dependent claim 23 depends on claim 22, and by virtue of this dependency, is nonobvious over the cited references for the reasons set forth above with respect to claim 22. Dependent claim 23 is also nonobvious due to the limitations recited therein. In particular, dependent claim 23 recites, *inter alia*, the following limitations (emphasis ours):

“wherein the instructions to maintain at least the server-side connection persistent include instructions executable by said processor to:

de-link, by said network device, the server-side connection from the client-side connection in response to a RESET packet received via the client-side connection.”

These limitations are not met by the cited references. Page 7 (section 15) of the final Office Action of June 12, 2008 has admitted that neither Yang nor Erickson teach de-linking the server-side connection from the client-side connection in response to a RESET packet. To supply the missing teachings of Yang and Erickson, the final Office Action of June 12, 2008 relies upon Kirby.

However, Kirby does not cure the deficiencies of Yang and Erickson.

Specifically, claim 23 requires “the instructions to maintain ... persistent include instructions ... to: de-link ... the server-side connection from the client-side connection in response to a RESET packet.” Kirby does not teach instructions to maintain persistent in response to a received RESET packet. Quite the opposite to claim 23, Kirby teaches terminating a connection in response to a RESET (RST) packet—Kirby is completely silent as to any relationship between his RST packet and instructions to maintain persistent as recited in claim 23.

For example, Kirby teaches the following in his column 4, lines 8-11 (emphasis ours):

“The connection packet handler 108 watches 120 (FIG. 7) packets passing on link 20 for those of types involved in setting up and terminating connections (e.g., SYN, FIN, and RST packets).”

From the above-quoted passage of Kirby, he teaches that his RST packet is “involved in setting up and terminating connections.” Accordingly, Kirby does not meet the limitations of claim 23 that require “instructions to maintain ... persistent ... in response to a RESET packet.”

In view of the above, claim 23 is not obvious.

T. Rejection of dependent claim 47 on the basis of Yang, Erickson, and Kirby

Dependent claim 47 depends on claim 22, and by virtue of this dependency, is nonobvious over the cited references for the reasons set forth above with respect to claim 22. Dependent claim 47 is also nonobvious due to the limitations recited therein. In particular, dependent claim 47 recites, *inter alia*, the following limitations (emphasis ours):

“wherein the instructions to maintain at least the server-side connection persistent ... include instructions executable by said processor to:

de-link, by said network device, the server-side connection from the client-side connection in response to a FIN packet received via the client-side connection.”

These limitations are not met by the cited references. Page 7 (section 15) of the final Office Action of June 12, 2008 has admitted that neither Yang nor Erickson teach de-linking the server-side connection from the client-side connection in response to a FIN packet. To supply the missing teachings of Yang and Erickson, the final Office Action of June 12, 2008 relies upon Kirby.

However, Kirby does not cure the deficiencies of Yang and Erickson.

Specifically, claim 47 requires “instructions to maintain ... persistent include instructions ... to: de-link ... the server-side connection from the client-side connection in response to a FIN packet.” Kirby does not teach instructions to maintain persistency in response to a FIN packet. Quite the opposite to claim 47, Kirby teaches terminating a connection in response to a FIN packet—Kirby is completely silent as to any relationship between his FIN packet and instructions to maintain persistency as recited in claim 47.

For example, Kirbys’ column 4, lines 59-67 (previously quoted above) teaches that his FIN packet is used “to delete the connection between device 12 and device 26.” Accordingly, Kirby does not meet the limitations of claim 47 that require “instructions to maintain ... persistent ... in response to a FIN packet.”

In view of the above, claim 47 is not obvious.

U. Rejection of dependent claim 25 on the basis of Yang and Erickson

Dependent claim 25 depends on claim 22, and by virtue of this dependency, is nonobvious over the cited references for the reasons set forth above with respect to claim 22. Dependent claim 25 is also nonobvious due to the limitations recited therein. In particular, dependent claim 25 recites, *inter alia*, the following limitations (emphasis ours):

“wherein the instructions to maintain at least the server-side connection persistent include instructions executable by said processor to: modify, by said network device, header information in the communication that signals the server-side connection to close to a format unrecognizable by a server coupled to said server-side connection, to cause the server to ignore the modified header information.”

These limitations are not met by the cited references. The final Office Action of June 12, 2008 has not cited any teaching of Yang that meets these limitations of claim 25. Instead, the final Office Action of June 12, 2008 relies upon column 6, line 28 to column 8, line 67 of Erickson, a portion of which includes column 6, line 58 to column 7, line 16 previously quoted above. In the passage of Erickson relied upon by the final Office Action of June 12, 2008, Erickson is silent as to any header information having “a format that signals the server-side connection to close” as recited in claim 25. Instead, Erickson teaches in column 6, line 58 to column 7, line 16 that the “MD_UPLOAD_READAHEAD_SIZE parameter” that is typically “set to a value that is an average size of a request message received,” and then modifying this parameter in order to have the web server “ignore processing of chunked data that is designated to be passed to the extension 132.” The “MD_UPLOAD_READAHEAD_SIZE parameter” of Erickson is not the same as the claimed header information having “a format that signals the server-side connection to close,” since the MD_UPLOAD_READAHEAD_SIZE parameter has to provide “a value that is an average size of a request message received,” thereby implicitly/explicitly teaching an open connection to receive/process the request message, rather than a “a format that signals the server-side connection to close” as required in claim 25.

Furthermore, Erickson's teaches in column 6, line 58 to column 7, line 16 (quoted previously above) that modification of the "MD_UPLOAD_READAHEAD_SIZE parameter" involves modification of a parameter in the web server 120 by the extension 132. In contrast, claim 25 requires modification of "header information in the communication." Modification of "header information in the communication" is not the same as Erickson's modification of a parameter in the web server 120.

In view of the above, claim 25 is not obvious.

V. Rejection of dependent claim 26 on the basis of Yang and Erickson

Dependent claim 26 depends on claim 22, and by virtue of this dependency, is nonobvious over the cited references for the reasons set forth above with respect to claim 22. Dependent claim 26 is also nonobvious due to the limitations recited therein. In particular, dependent claim 26 recites, *inter alia*, the following limitations (emphasis ours):

"wherein the instructions to maintain at least the server-side connection persistent include instructions executable by said processor to:
modify, by said network device, a protocol version number indicated in the communication to a different protocol version number that corresponds to a persistent connection."

These limitations are not met by the cited references. The final Office Action of June 12, 2008 has not cited any teaching of Yang that meets these limitations of claim 26. Instead, the final Office Action of June 12, 2008 relies upon column 8, line 19 to column 9, line 38 of Erickson, a portion of which includes column 8, lines 55-58 previously quoted above. This passage of Erickson relied upon by the final Office Action of June 12, 2008 is silent as to any instructions to change a protocol version number as recited in claim 26. Instead, Erickson teaches in his column 8, lines 55-58 that his "invention is only concerned with the content of the fields" of the HTTP/1.1 specification. Erickson provides absolutely no teaching with respect to changing a protocol version number (*e.g.*, an HTTP version number other than HTTP 1.1.) indicated in his fields to another HTTP version number, such as HTTP 1.1.

In view of the above, claim 26 is not obvious.

W. Rejection of independent claim 27 and dependent claim 48 on the basis of Yang and Erickson

Independent claim 27 recites, *inter alia*, a network device having (emphasis ours):

“means for ... extending persistency of the server-side connection if the read content of the packet signals said server-side connection to close.”

Page 3 (section 5) of the final Office Action of June 12, 2008 has admitted that Yang does not teach “receiving at the network device via the client-side connection a communication that signals the server-side connection to close.”

As explained on pages 13-14 of the amendment of March 5, 2008, Yang’s column 2, lines 21-41; column 5, lines 51-53; and column 8, lines 17-19 teach a technique to create a quasi-persistent connection “by modifying the content-length key-value pair in a HTTP request or response” so that “the content-length key-value pair has a content-length value that is large”—in this manner, the “client or server that receives the content-length key-value pair having the large content-length value, in effect, believes that a large amount of data will be included with the request or response,” and therefore does not close the connection.

It is therefore not possible for Yang to meet the limitations of claim 27 that require “means for ... extending persistency of the server-side connection if the read content ... signals said server-side connection to close.” For instance, if Yang’s client modified the content-length key-value pair to extend persistency as taught by Yang, and then this modified content-length key-value pair is sent by Yang’s client to his server, the received modified content-length key-value pair would signal his server-side connection to remain open, rather than “signals said server-side connection to close” as required by claim 27.

To supply the missing teachings of Yang, page 7 (section 16) of the final Office Action of June 12, 2008 relies upon Erickson. However, Erickson does not cure the deficiencies of Yang.

Erickson's invention is directed towards HTTP version 1.1 communications, as Erickson explicitly teaches in his column 8, lines 19-34 (previously quoted above). As abundantly clear from the above-quoted passage of column 8, lines 19-34 from Erickson, his invention is directed towards and is limited to an HTTP 1.1 implementation. As described in pages 1-4 of the present application, HTTP 1.1 provides persistent connections as a default, and Erickson's column 2, lines 29-30 also concurs that HTTP 1.1 provides a persistent connection ("the newer HTTP 1.1 specification provides a keep-alive mechanism that allows one connection for multiple objects on an HTML page"). Accordingly, the HTTP 1.1 implementation of Erickson's invention does not in and of itself address "content of the packet [that] signals said server-side connection to close" of claim 27.

Rather, Erickson's invention provides a persistency technique to addresses the situation in an HTTP 1.1 implementation where a connection might potentially be closed "after a period of inactivity." In his column 2, lines 27-32 and column 9, lines 1-32 (previously quoted above), Erickson teaches rather clearly that his invention is directed towards an HTTP 1.1 implementation, in particular use of the "chunking" aspect of HTTP 1.1. As taught by Erickson, his invention "generates a chunk with only the Keep-Alive data field" and then sends this chunk having the Keep-Alive data, which "allows the connection to remain alive even during periods of inactivity."

At least two things can be readily gleaned from Erickson's teachings in his column 2, lines 27-32 and column 9, lines 1-32. First, his chunk having the "Keep-Alive" data can in no way meet the limitations of claim 27 that require "content of a packet [that] signals said server-side connection to close." The packet content of claim 27 "signals the server-side connection to close," whereas Erickson's chunk (having the "Keep Alive" data) signals his "connection to remain alive." These are opposite/contradictory teachings.

Second, Erickson's invention addresses the situation where a connection might close "if additional chunks are not received within a certain time frame" and "during periods of inactivity." It is implicit/explicit therefore, that there is no communications of any sort (in particular "content of the packet [that] signals said server-side connection to close") that is being sent/received during these inactivity periods or time frame of Erickson. Thus, Erickson again does not meet the limitations of claim 27 that require "means for extending persistency ... if the

read content of the packet signals said server-side connection to close,” since Erickson is addressing a situation where there is no communication (“periods of inactivity”) at all and therefore has to provide the chunk having the Keep-Alive data in order to prevent the connection from closing due to inactivity.

Page 3 (section 5) of the final Office Action cites Erickson’s column 2, lines 22-39 as allegedly disclosing “a communication that signals the server-side connection to close.” However, a thorough and accurate reading of this passage reveals that Erickson is merely discussing (in column 2, lines 22-27) the HTTP versions prior to HTTP 1.1. In these earlier versions prior to HTTP 1.1 (and as also explained in pages 1-4 of the present application), the connection is closed after the response is sent. This cited passage of Erickson, while speaking of closing of a connection after sending a response, is not applicable to and has nothing to do with his invention. As previously explained above, Erickson’s invention is explicitly directed towards (and limited to) HTTP 1.1 that uses chunking and wherein the default behavior is to keep a connection persistent, except after a period of inactivity—Erickson teaches an invention to maintain the connection persistent to prevent closure due to a period of inactivity in an HTTP 1.1 implementation—Erickson is completely silent as to any applicability and as to any possible manner of implementation of his invention with respect to other HTTP versions (per his column 2, lines 22-27) that existed prior to HTTP 1.1 and that closed a connection after sending a response. Thus, the passage in column 2, lines 22-27 of Erickson discussing HTTP versions that were prior to HTTP 1.1, considered as a whole along with his other teachings, does not meet the limitations of claim 27 that require “means for ... extending persistency ... if the read content of the packet signals said server-side connection to close.”

In view of the above, since all of the limitations of claim 27 are not met by the cited references, claim 27 is not obvious.

Dependent claim 48 depends on claim 27, and by virtue of this dependency, is nonobvious over the cited references for the reasons set forth above with respect to claim 27.

X. Rejection of dependent claim 28 on the basis of Yang and Erickson

Dependent claim 28 depends on claim 27, and by virtue of this dependency, is nonobvious over the cited references for the reasons set forth above with respect to claim 27.

Dependent claim 28 is also nonobvious due to the limitations recited therein. In particular, dependent claim 28 recites, *inter alia*, the following limitations (emphasis ours):

“wherein the means for extending the persistency of the server-side connection modifies header information in the packet to a format unrecognizable to a server coupled to said server-side connection to receive said packet.”

These limitations are not met by the cited references. The final Office Action of June 12, 2008 has not cited any teaching of Yang that meets these limitations of claim 28. Instead, the final Office Action of June 12, 2008 relies upon column 6, line 28 to column 8, line 67 of Erickson, a portion of which includes column 6, line 58 to column 7, line 16 previously quoted above. In this passage of Erickson relied upon by the final Office Action of June 12, 2008, Erickson teaches that modification of the “MD_UPLOAD_READAHEAD_SIZE parameter” involves modification of a parameter in the web server 120 by the extension 132. In contrast, claim 28 requires modification of “header information in the packet.” Modification of “header in the packet” is not the same as Erickson’s modification of a parameter in the web server 120.

In view of the above, claim 28 is not obvious.

Y. Rejection of dependent claim 29 on the basis of Yang and Erickson

Dependent claim 29 depends on claim 27, and by virtue of this dependency, is nonobvious over the cited references for the reasons set forth above with respect to claim 27. Dependent claim 29 is also nonobvious due to the limitations recited therein. In particular, dependent claim 29 recites, *inter alia*, the following limitations (emphasis ours):

“wherein the means for extending the persistency of the server-side connection modifies header information in the packet to indicate a protocol version that corresponds to a persistent connection.”

These limitations are not met by the cited references. The final Office Action of June 12, 2008 has not cited any teaching of Yang that meets these limitations of claim 29. Instead, the final Office Action of June 12, 2008 relies upon column 8, line 19 to column 9, line 38 of Erickson, a portion of which includes column 8, lines 55-58 previously reproduced above. This passage of Erickson relied upon by the final Office Action of June 12, 2008 is silent as to any modification of header information in a packet to indicate a protocol version that corresponds to a persistent connection as recited in claim 29. Instead, Erickson teaches in his column 8, lines 55-58 that his “invention is only concerned with the content of the fields” of the HTTP/1.1 specification, which makes implicit/explicit that Erickson relates only to HTTP 1.1 and does not perform any modification of non-HTTP 1.1 header information to HTTP 1.1 header information.

In view of the above, claim 29 is not obvious.

Z. Rejection of dependent claims 30 and 50 on the basis of Yang, Erickson, and Kirby

Dependent claims 30 and 50 depend on claim 27, and by virtue of this dependency, are nonobvious over the cited references for the reasons set forth above with respect to claim 27. Dependent claims 30 and 50 are also nonobvious due to the limitations recited therein. In particular, dependent claims 30 and 50 recite, *inter alia* and using varying language, the following limitations (emphasis ours):

“wherein the means for extending the persistency ... de-links the server-side connection from the client-side connection in response to RESET content in the packet.”

These limitations are not met by the cited references. Page 7 (section 15) of the final Office Action of June 12, 2008 has admitted that neither Yang nor Erickson teach de-linking the server-side connection from the client-side connection, in response to RESET content in the packet. To supply the missing teachings of Yang and Erickson, the final Office Action of June 12, 2008 relies upon Kirby.

However, Kirby does not cure the deficiencies of Yang and Erickson.

Specifically, claims 30 and 50 require “extending the persistency ... in response to RESET content in the packet.” Kirby does not teach extending persistency if a received packet contains RESET content. Quite the opposite to claims 30 and 50, Kirby teaches terminating a connection in response to a RESET (RST) packet—Kirby is completely silent as to any relationship between his RST packet and extending persistency as recited in claims 30 and 50.

For example, Kirby teaches the following in his column 4, lines 8-11 (emphasis ours):

“The connection packet handler 108 watches 120 (FIG. 7) packets passing on link 20 for those of types involved in setting up and terminating connections (e.g., SYN, FIN, and RST packets).”

From the above-quoted passage of Kirby, he teaches that his RST packet is “involved in setting up and terminating connections.” Accordingly, Kirby does not meet the limitations of claims 30 and 50 that require extending persistency in response to RESET content in the packet.

In view of the above, claims 30 and 50 are not obvious.

AA. Rejection of dependent claim 49 on the basis of Yang, Erickson, and Kirby

Dependent claim 49 depends on claim 27, and by virtue of this dependency, is nonobvious over the cited references for the reasons set forth above with respect to claim 27. Dependent claim 49 is also nonobvious due to the limitations recited therein. In particular, dependent claim 49 recites, *inter alia*, the following limitations (emphasis ours):

“wherein the means for extending the persistency of the server-side connection ... maintains persistency by de-linking the server-side connection from the client-side connection in response to FIN content in the packet.”

These limitations are not met by the cited references. Page 7 (section 15) of the final Office Action of June 12, 2008 has admitted that neither Yang nor Erickson teach de-linking the server-side connection from the client-side connection, in response to FIN content in the packet. To supply the missing teachings of Yang and Erickson, the final Office Action of June 12, 2008 relies upon Kirby.

However, Kirby does not cure the deficiencies of Yang and Erickson.

Specifically, claim 49 requires “maintains persistency ... in response to FIN content in the packet.” Kirby does not teach maintaining persistency in response to FIN content in a packet. Quite the opposite to claim 49, Kirby teaches terminating a connection in response to a FIN packet—Kirby is completely silent as to any relationship between his FIN packet and maintaining persistency as recited in claim 49.

For example, Kirbys’ column 4, lines 59-67 (previously quoted above) teaches that his FIN packet is used “to delete the connection between device 12 and device 26.” Accordingly, Kirby does not meet the limitations of claim 49 that require “maintains persistency ... in response to FIN content in the packet.”

In view of the above, claim 49 is not obvious.

AB. Rejection of dependent claim 51 on the basis of Yang and Erickson

Dependent claim 51 depends on claim 27, and by virtue of this dependency, is nonobvious over the cited references for the reasons set forth above with respect to claim 27. Dependent claim 51 is also nonobvious due to the limitations recited therein. In particular, dependent claim 51 recites, *inter alia*, the following limitations (emphasis ours):

“wherein the means for extending the persistency of the server-side connection, if the read content of the packet signals said server-side connection to close, maintains persistency by replacing a Connection: Close header of the packet with a Connection: Keep-Alive header.”

These limitations are not met by the cited references. The final Office Action of June 12, 2008 has not cited any teaching of Yang that meets these limitations of claim 51. Instead, the final Office Action of June 12, 2008 relies upon column 7, line 16 to column 8, line 67 of Erickson, a portion of which includes column 8, lines 19-54 that was previously quoted above.

Erickson is silent as to any replacing of “a Connection: Close header of the packet with a Connection: Keep-Alive header” as recited in claim 51. Instead, column 8, lines 19-54 of Erickson merely teaches using Keep-Alive data in order to keep his connection alive, where the connection might otherwise be closed due to “periods of idleness.” “Periods of idleness” where no communications are received are not the same as the claimed packet that is received and that signals the server-side connection with close (with a Connection: Close header) of claim 51.

In view of the above, claim 51 is not obvious.

AC. Rejection of independent claim 31 on the basis of Yang and Erickson

Independent claim 31 recites, *inter alia*, the following limitations (emphasis ours):

“software executable by the processor to maintain persistent at least the server-side connection in response to a communication received via said client-side connection that signals said server-side connection to close.”

Page 3 (section 5) of the final Office Action of June 12, 2008 has admitted that Yang does not teach “receiving at the network device via the client-side connection a communication that signals the server-side connection to close.”

As explained on pages 13-14 of the amendment of March 5, 2008, Yang’s column 2, lines 21-41; column 5, lines 51-53; and column 8, lines 17-19 teach a technique to create a quasi-persistent connection “by modifying the content-length key-value pair in a HTTP request or response” so that “the content-length key-value pair has a content-length value that is large”—in this manner, the “client or server that receives the content-length key-value pair having the large content-length value, in effect, believes that a large amount of data will be included with the request or response,” and therefore does not close the connection.

It is therefore not possible for Yang to meet the limitations of claim 31 that require that the server-side connection is maintained persistent “in response to a communication received via said client-side connection that signals said server-side connection to close.” For instance, if Yang’s client modified the content-length key-value pair to provide persistency as taught by Yang, and then this modified content-length key-value pair is sent by Yang’s client to his server, the received modified content-length key-value pair would signal his server-side connection to remain open, rather than “signals said server-side connection to close” as required by claim 31.

To supply the missing teachings of Yang, the final Office Action of June 12, 2008 relies upon Erickson. However, Erickson does not cure the deficiencies of Yang.

Erickson’s invention is directed towards and limited to an HTTP version 1.1 implementation, as Erickson explicitly teaches in his column 8, lines 19-34 (previously quoted above). As described in pages 1-4 of the present application, HTTP 1.1 provides persistent connections as a default, and Erickson’s column 2, lines 29-30 also concurs that HTTP 1.1 provides a persistent connection (“the newer HTTP 1.1 specification provides a keep-alive mechanism that allows one connection for multiple objects on an HTML page”). Accordingly, the HTTP 1.1 implementation of Erickson’s invention does not in and of itself address “a communication ... that signals said server-side connection to close” of claim 31.

Rather, Erickson’s invention provides a persistency technique to addresses the situation in an HTTP 1.1 implementation where a connection might potentially be closed “after a period of inactivity.” In his column 2, lines 27-32 and column 9, lines 1-32 (previously quoted above), Erickson teaches rather clearly that his invention is directed towards an HTTP 1.1 implementation, in particular use of the “chunking” aspect of HTTP 1.1. As taught by Erickson, his invention “generates a chunk with only the Keep-Alive data field” and then sends this chunk having the Keep-Alive data, which “allows the connection to remain alive even during periods of inactivity.”

At least two things can be readily gleaned from Erickson’s teachings in his column 2, lines 27-32 and column 9, lines 1-32. First, his chunk having the “Keep-Alive” data can in no way meet the limitations of claim 31 that require “a communication ... that signals said server-side connection to close.” The communication of claim 31 “signals the server-side

connection to close,” whereas Erickson’s chunk (having the “Keep Alive” data) signals his “connection to remain alive.” These are opposite/contradictory teachings.

Second, Erickson’s invention addresses the situation where a connection might close “if additional chunks are not received within a certain time frame” and “during periods of inactivity.” It is implicit/explicit therefore, that there is no communications of any sort (in particular “a communication that signals said server-side connection to close”) that is being sent/received during these inactivity periods or time frame of Erickson. Thus, Erickson again does not meet the limitations of claim 31 that require “maintain persistent ... in response to a communication ... that signals said server-side connection to close,” since Erickson is addressing a situation where there is no communication (“periods of inactivity”) at all and therefore has to provide the chunk having the Keep-Alive data in order to prevent the connection from closing due to inactivity.

Page 3 (section 5) of the final Office Action cites Erickson’s column 2, lines 22-39 as allegedly disclosing “a communication that signals the server-side connection to close.” However, a thorough and accurate reading of this passage reveals that Erickson is merely discussing (in column 2, lines 22-27) the HTTP versions prior to HTTP 1.1. In these earlier versions prior to HTTP 1.1 (and as also explained in pages 1-4 of the present application), the connection is closed after the response is sent. This cited passage of Erickson, while speaking of closing of a connection after sending a response, is not applicable to and has nothing to do with his invention. As previously explained above, Erickson’s invention is explicitly directed towards (and limited to) HTTP 1.1 that uses chunking and wherein the default behavior is to keep a connection persistent, except after a period of inactivity—Erickson teaches an invention to maintain the connection persistent to prevent closure due to a period of inactivity in an HTTP 1.1. implementation—Erickson is completely silent as to any applicability and as to any possible manner of implementation of his invention with respect to other HTTP versions (per his column 2, lines 22-27) that existed prior to HTTP 1.1 and that closed a connection after sending a response. Thus, the passage in column 2, lines 22-27 of Erickson discussing HTTP versions that were prior to HTTP 1.1, considered as a whole along with his other teachings, does not meet the limitations of claim 31 that require “maintain persistent ... in response to a communication ... that signals said server-side connection to close.”

In view of the above, since all of the limitations of claim 31 are not met by the cited references, claim 31 is not obvious.

AD. Rejection of dependent claim 32 on the basis of Yang and Erickson

Dependent claim 32 depends on claim 31, and by virtue of this dependency, is nonobvious over the cited references for the reasons set forth above with respect to claim 31. Dependent claim 32 is also nonobvious due to the limitations recited therein. In particular, dependent claim 32 recites, *inter alia*, the following limitations (emphasis ours):

“wherein the software executable by said processor of said network device to maintain the server-side connection persistent includes code to modify a format of header information in the communication received via the client-side connection from a format that signals said server-side connection to close to a format that is unrecognizable by a server coupled to said server-side connection, to cause the server to ignore the header information.”

These limitations are not met by the cited references. The final Office Action of June 12, 2008 has not cited any teaching of Yang that meets these limitations of claim 32. Instead, the final Office Action of June 12, 2008 relies upon column 6, line 28 to column 8, line 67 of Erickson, a portion of which includes column 6, line 58 to column 7, line 16 previously quoted above. In the passage of Erickson relied upon by the final Office Action of June 12, 2008, Erickson is silent as to any header information having “a format that signals the server-side connection to close” as recited in claim 32. Instead, Erickson teaches in column 6, line 58 to column 7, line 16 that the “MD_UPLOAD_READAHEAD_SIZE parameter” that is typically “set to a value that is an average size of a request message received,” and then modifying this parameter in order to have the web server “ignore processing of chunked data that is designated to be passed to the extension 132.” The “MD_UPLOAD_READAHEAD_SIZE parameter” of Erickson is not the same as the claimed header information having “a format that signals the server-side connection to close,” since the MD_UPLOAD_READAHEAD_SIZE parameter has

to provide “a value that is an average size of a request message received,” thereby implicitly/explicitly teaching an open connection to receive/process the request message, rather than a “a format that signals the server-side connection to close” as required in claim 32.

Furthermore, Erickson’s teaches in column 6, line 58 to column 7, line 16 (quoted previously above) that modification of the “MD_UPLOAD_READAHEAD_SIZE parameter” involves modification of a parameter in the web server 120 by the extension 132. In contrast, claim 32 requires modification of “header information in the communication.” Modification of “header information in the communication” is not the same as Erickson’s modification of a parameter in the web server 120.

In view of the above, claim 32 is not obvious.

AE. Rejection of dependent claim 33 on the basis of Yang and Erickson

Dependent claim 33 depends on claim 31, and by virtue of this dependency, is nonobvious over the cited references for the reasons set forth above with respect to claim 31. Dependent claim 33 is also nonobvious due to the limitations recited therein. In particular, dependent claim 33 recites, *inter alia*, the following limitations (emphasis ours):

“wherein the software executable by said processor of said network device to maintain the server-side connection persistent includes code to modify a HTTP protocol version value indicated in the communication received via the client-side connection to a HTTP protocol version value that is associated with a persistent connection.”

These limitations are not met by the cited references. The final Office Action of June 12, 2008 has not cited any teaching of Yang that meets these limitations of claim 33. Instead, the final Office Action of June 12, 2008 relies upon column 8, line 19 to column 9, line 38 of Erickson, a portion of which includes column 8, lines 55-58 previously quoted above. This passage of Erickson relied upon by the final Office Action of June 12, 2008 is silent as to any code to modify an HTTP protocol version value as recited in claim 33. Instead, Erickson teaches in his column 8, lines 55-58 that his “invention is only concerned with the content of the fields”

of the HTTP/1.1 specification. Erickson provides absolutely no teaching with respect to modification of an HTTP protocol version value (*e.g.*, an HTTP version value other than HTTP 1.1.) indicated in his fields to another HTTP version value, such as HTTP 1.1.

In view of the above, claim 33 is not obvious.

AF. Rejection of dependent claim 34 on the basis of Yang, Erickson, and Kirby

Dependent claim 34 depends on claim 31, and by virtue of this dependency, is nonobvious over the cited references for the reasons set forth above with respect to claim 31. Dependent claim 34 is also nonobvious due to the limitations recited therein. In particular, dependent claim 34 recites, *inter alia*, the following limitations (emphasis ours):

“wherein the software executable by said processor of said network device to maintain the server-side connection persistent includes code to de-link the server-side connection from the client-side connection in response to said communication, containing a RESET, received via said client-side connection.”

These limitations are not met by the cited references. Page 7 (section 15) of the final Office Action of June 12, 2008 has admitted that neither Yang nor Erickson teach de-linking the server-side connection from the client-side connection in response to a communication containing a RESET. To supply the missing teachings of Yang and Erickson, the final Office Action of June 12, 2008 relies upon Kirby.

However, Kirby does not cure the deficiencies of Yang and Erickson.

Specifically, claim 34 requires “software executable by said processor of said network device to maintain the server-side connection persistent ... in response to said communication, containing a RESET.” Kirby does not teach software to maintain a connection persistent in response to a communication containing a RESET. Quite the opposite to claim 34, Kirby teaches terminating a connection in response to a RESET (RST) packet—Kirby is

completely silent as to any relationship between his RST packet and software to maintain a connection persistent as recited in claim 34.

For example, Kirby teaches the following in his column 4, lines 8-11 (emphasis ours):

“The connection packet handler 108 watches 120 (FIG. 7) packets passing on link 20 for those of types involved in setting up and terminating connections (e.g., SYN, FIN, and RST packets).”

From the above-quoted passage of Kirby, he teaches that his RST packet is “involved in setting up and terminating connections.” Accordingly, Kirby does not meet the limitations of claim 34 that require “maintain ... persistent ... in response to said communication, containing a RESET.”

In view of the above, claim 34 is not obvious.

AG. Rejection of dependent claim 35 on the basis of Yang and Erickson

Dependent claim 35 depends on claim 31, and by virtue of this dependency, is nonobvious over the cited references for the reasons set forth above with respect to claim 31. Dependent claim 35 is also nonobvious due to the limitations recited therein. In particular, dependent claim 35 recites, *inter alia*, the following limitations (emphasis ours):

“wherein the software executable by said processor of said network device to maintain the server-side connection persistent includes code to modify a Connection: Close header in the communication to Connection: Keep-Alive header.”

These limitations are not met by the cited references. The final Office Action of June 12, 2008 has not cited any teaching of Yang that meets these limitations of claim 35. Instead, the final Office Action of June 12, 2008 relies upon column 7, line 16 to column 8, line 67 of Erickson, a portion of which from column 8, lines 19-54 was previously quoted above.

Erickson is silent as to any modification of “a Connection: Close header of the communication to a Connection: Keep-Alive header” as recited in claim 35. Instead, column 8, lines 19-54 of Erickson merely teaches using Keep-Alive data in order to keep his connection alive, where the connection might otherwise be closed due to “periods of idleness.” “Periods of idleness” where no communications are received are not the same as the claimed communication that is received and that signals the server-side connection with close (with the Connection: Close header) of claim 35.

In view of the above, claim 35 is not obvious.

AH. Rejection of dependent claim 36 on the basis of Yang, Erickson, and Kirby

Dependent claim 36 depends on claim 31, and by virtue of this dependency, is nonobvious over the cited references for the reasons set forth above with respect to claim 31.

AI. Rejection of dependent claim 52 on the basis of Yang, Erickson, and Kirby

Dependent claim 52 depends on claim 31, and by virtue of this dependency, is nonobvious over the cited references for the reasons set forth above with respect to claim 31. Dependent claim 52 is also nonobvious due to the limitations recited therein. In particular, dependent claim 52 recites, *inter alia*, the following limitations (emphasis ours):

“wherein the software executable by said processor of said network device to maintain the server-side connection persistent, in response to said communication received via said client-side connection that signals said server-side connection to close, includes code to de-link the server-side connection from the client-side connection in response to said communication, containing a FIN, received via said client-side connection.”

These limitations are not met by the cited references. Page 7 (section 15) of the final Office Action of June 12, 2008 has admitted that neither Yang nor Erickson teach de-linking the server-side connection from the client-side connection in response to a communication containing a FIN. To supply the missing teachings of Yang and Erickson, the final Office Action of June 12, 2008 relies upon Kirby.

However, Kirby does not cure the deficiencies of Yang and Erickson.

Specifically, claim 52 requires “maintain ... persistent ... in response to said communication, containing a FIN.” Kirby does not teach instructions to maintain persistency in response to a communication containing a FIN. Quite the opposite to claim 52, Kirby teaches terminating a connection in response to a FIN packet—Kirby is completely silent as to any relationship between his FIN packet and code to maintain persistency as recited in claim 52.

For example, Kirbys’ column 4, lines 59-67 (previously quoted above) teaches that his FIN packet is used “to delete the connection between device 12 and device 26.” Accordingly, Kirby does not meet the limitations of claim 52 that require “maintain ... persistent ... in response to said communication, containing a FIN.”

In view of the above, claim 52 is not obvious.

In view of the arguments as set forth above, the Examiner’s rejections of the claims should be withdrawn.

Respectfully submitted,
SEED Intellectual Property Law Group PLLC

/Dennis M. de Guzman/

Dennis M. de Guzman
Registration No. 41,702

DMD:wt

701 Fifth Avenue, Suite 5400
Seattle, Washington 98104
Phone: (206) 622-4900
Fax: (206) 682-6031

1237500_1.DOC

VIII. CLAIMS APPENDIX

1. A method, comprising:
at a network device, providing a hypertext transfer protocol (HTTP) client-side connection and a HTTP server-side connection;
receiving at said network device via said client-side connection a communication that signals said server-side connection to close; and
maintaining persistent, by said network device, at least the server-side connection in response to said communication received via said client-side connection.
2. The method of claim 1, further comprising closing the client-side connection while the server-side connection is maintained persistent.
3. The method of claim 1 wherein maintaining persistent at least the server-side connection in response to said communication received via said client-side connection includes:
de-linking, by said network device, the server-side connection from the client-side connection in response to a RESET packet received by said network device via said client-side connection.
4. The method of claim 1 wherein maintaining persistent at least the server-side connection in response to said communication received via said client-side connection includes:
de-linking, by said network device, the server-side connection from the client-side connection in response to a FIN packet received by said network device via said client-side connection.

5. The method of claim 1, further comprising said network device closing both the client-side and server-side connections in response to a FIN packet received by said network device via said server-side connection.

6. The method of claim 1 wherein maintaining persistent at least the server-side connection in response to said communication received via said client-side connection includes:

identifying, by said network device, a Connection: Close header in the communication received via the client-side connection; and

replacing, by said network device, the Connection: Close header in the communication with a Connection: Keep-Alive header.

7. The method of claim 6, further comprising said network device performing at least one of increasing a total length of a packet having the Connection: Close header, fragmenting the packet having the Connection: Close header, and recalculating a checksum of the packet.

8. The method of claim 1 wherein maintaining persistent at least the server-side connection in response to said communication received via said client-side connection includes:

inserting, by said network device, a Connection: Keep-Alive header in the communication if the communication does not contain any header information indicative of whether to close the HTTP connection.

9. The method of claim 1 wherein maintaining persistent at least the server-side connection in response to said communication received via said client-side connection includes:

modifying, by said network device, a header in the communication from a format that signals the server-side connection to close to a format that is unrecognizable by a server coupled to said server-side connection, to cause the server to ignore the modified header.

10. The method of claim 9 wherein modifying the header in the request to the form that is unrecognizable to the server includes at least one of modifying a name of the header and modifying a value of the header.

11. The method of claim 1 wherein maintaining persistent at least the server-side connection in response to said communication received via said client-side connection includes:

changing, by said network device, a HTTP version value indicated in the communication to another HTTP version value that is recognizable by a server, coupled to said server-side connection, as being associated with a persistent connection.

12. The method of claim 11, further comprising adjusting a checksum based on a difference between the HTTP version values.

13. The method of claim 1 wherein the communication includes a header having a proxy format.

14. A method, comprising:
establishing at a network device a client-side connection and a server-side connection;
reading, by said network device, content of a packet received via the client-side connection; and
extending, by said network device, persistency of the server-side connection if the read content of the packet signals said server-side connection to close.

15. The method of claim 14 wherein establishing the client-side and server-side connections include establishing these connections as part of a hypertext transfer protocol (HTTP) connection.

16. The method of claim 14 wherein extending, by said network device, the persistency of the server-side connection includes:

de-linking, by said network device, the server-side connection from the client-side connection, if the read content indicates that the packet is a RESET packet received via the client-side connection.

17. The method of claim 14 wherein extending, by said network device, the persistency of the server-side connection includes:

de-linking, by said network device, the server-side connection from the client-side connection, if the read content indicates that the packet is a FIN packet received via the client-side connection.

18. The method of claim 14 wherein extending, by said network device, the persistency of the server-side connection includes:

identifying, by said network device, header information of the packet received via the client-side connection that signals the server-side connection to close; and

replacing, by said network device, the identified header information with new header information that maintains the server-side connection persistent.

19. The method of claim 14 wherein extending, by said network device, the persistency of the server-side connection includes:

determining, by said network device, that header information of the packet received via the client-side connection does not include any information that signals the server-side connection to close; and

applying, by said network device, header information in the packet that maintains the server-side connection persistent.

20. The method of claim 14 wherein extending, by said network device, the persistency of the server-side connection includes:

modifying, by said network device, header information in the packet received via the client-side connection from a format that signals said server-side connection to close to a format that is unrecognizable by a server that is to receive the packet via the server-side connection.

21. The method of claim 14 wherein extending, by said network device, the persistency of the server-side connection includes:

changing, by said network device, a protocol version value indicated in the packet received via the client-side connection to a different protocol version value that corresponds to maintaining a persistent connection.

22. An article of manufacture, comprising:

a storage medium having instructions stored thereon that are executable by a processor of a network device to:

provide at said network device a hypertext transfer protocol (HTTP) client-side connection and a HTTP server-side connection;

receive via said client-side connection a communication that signals said server-side connection to close; and

maintain persistent, by said network device, at least the server-side connection in response to said communication received via said client-side connection.

23. The article of manufacture of claim 22 wherein the instructions to maintain at least the server-side connection persistent include instructions executable by said processor to:

de-link, by said network device, the server-side connection from the client-side connection in response to a RESET packet received via the client-side connection.

24. The article of manufacture of claim 22 wherein the instructions to maintain at least the server-side connection persistent include instructions executable by said processor to:

change, by said network device, header information in the communication received via the client-side connection to new header information corresponding to a persistent connection.

25. The article of manufacture of claim 22 wherein the instructions to maintain at least the server-side connection persistent include instructions executable by said processor to:

modify, by said network device, header information in the communication that signals the server-side connection to close to a format unrecognizable by a server coupled to said server-side connection, to cause the server to ignore the modified header information.

26. The article of manufacture of claim 22 wherein the instructions to maintain at least the server-side connection persistent include instructions executable by said processor to:

modify, by said network device, a protocol version number indicated in the communication to a different protocol version number that corresponds to a persistent connection.

27. An apparatus, comprising:

a network device adapted to be communicatively coupled between a client terminal and a server, the network device having:

communication terminal means for establishing a client-side connection and a server-side connection; and

means for reading content of a packet received via the client-side connection, and for extending persistency of the server-side connection if the read content of the packet signals said server-side connection to close.

28. The apparatus of claim 27 wherein the means for extending the persistency of the server-side connection modifies header information in the packet to a format unrecognizable to a server coupled to said server-side connection to receive said packet.

29. The apparatus of claim 27 wherein the means for extending the persistency of the server-side connection modifies header information in the packet to indicate a protocol version that corresponds to a persistent connection.

30. The apparatus of claim 27 wherein the means for extending the persistency of the server-side connection de-links the server-side connection from the client-side connection in response to RESET content in the packet received via the client side connection.

31. An apparatus, comprising:

a network device having:

first and second communication terminals, the first terminal being associated with a hypertext transfer protocol (HTTP) client-side connection and the second terminal being associated with a HTTP server-side connection;

a processor coupled to the first and second communication terminals; and

software executable by the processor to maintain persistent at least the server-side connection in response to a communication received via said client-side connection that signals said server-side connection to close.

32. The apparatus of claim 31 wherein the software executable by said processor of said network device to maintain the server-side connection persistent includes code to modify a format of header information in the communication received via the client-side connection from a format that signals said server-side connection to close to a format that is unrecognizable by a server coupled to said server-side connection, to cause the server to ignore the header information .

33. The apparatus of claim 31 wherein the software executable by said processor of said network device to maintain the server-side connection persistent includes code to modify a HTTP protocol version value indicated in the communication received via the client-side connection to a HTTP protocol version value that is associated with a persistent connection.

34. The apparatus of claim 31 wherein the software executable by said processor of said network device to maintain the server-side connection persistent includes code to de-link the server-side connection from the client-side connection in response to said communication, containing a RESET, received via said client-side connection.

35. The apparatus of claim 31 wherein the software executable by said processor of said network device to maintain the server-side connection persistent includes code to modify a Connection: Close header in the communication to Connection: Keep-Alive header.

36. The apparatus of claim 31 wherein the software is also executable by said processor of said network device to close the server-side connection in response to at least one of a RESET and FIN received via the server-side connection.

43. The method of claim 1 wherein said network device includes a switch.

44. The method of claim 11 wherein changing the HTTP version value to another HTTP version value includes changing, by said network device, from HTTP version 1.0 indicated in said request to HTTP version 1.1.

45. The method of claim 14 wherein said network device includes a switch.

46. The article of manufacture of claim 22 wherein said network device includes a switch.

47. The article of manufacture of claim 22 wherein the instructions to maintain at least the server-side connection persistent, in response to said communication received via said client-side connection, include instructions executable by said processor to:

de-link, by said network device, the server-side connection from the client-side connection in response to a FIN packet received via the client-side connection.

48. The apparatus of claim 27 wherein said network device includes a switch.

49. The apparatus of claim 27 wherein the means for extending the persistency of the server-side connection, if the read content of the packet signals said server-side connection

to close, maintains persistency by de-linking the server-side connection from the client-side connection in response to FIN content in the packet received via the client side connection.

50. The apparatus of claim 27 wherein the means for extending the persistency of the server-side connection, if the read content of the packet signals said server-side connection to close, maintains persistency by de-linking the server-side connection from the client-side connection in response to RESET content in the packet received via the client side connection.

51. The apparatus of claim 27 wherein the means for extending the persistency of the server-side connection, if the read content of the packet signals said server-side connection to close, maintains persistency by replacing a Connection: Close header of the packet with a Connection: Keep-Alive header.

52. The apparatus of claim 31 wherein the software executable by said processor of said network device to maintain the server-side connection persistent, in response to said communication received via said client-side connection that signals said server-side connection to close, includes code to de-link the server-side connection from the client-side connection in response to said communication, containing a FIN, received via said client-side connection.

IX. EVIDENCE APPENDIX

None.

X. RELATED PROCEEDINGS APPENDIX

None.